Stephen A. Wight
Klarquist Sparkman, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon 97204
Telephone: 503/226-7391

Our Ref. No. 3382-64896

For: ACCESS DRIVEN FILTERING
Express Mail Label EV339201001US Mailed: August 27, 2003 Sheet 1 of 17

FIG. 1

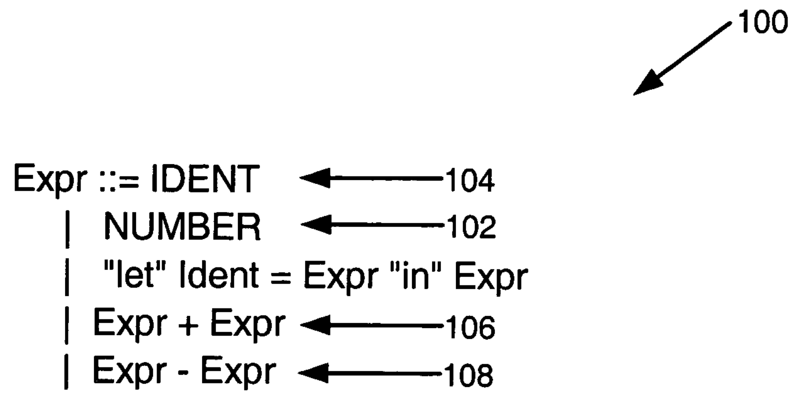
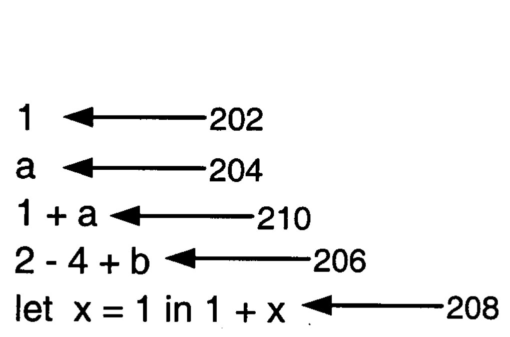


FIG. 2



Stephen A. Wight
Klarquist Sparkman, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon 97204
Telephone: 503/226-7391
For: ACCESS DRIVEN FILTERING
Express Mail Label EV339201001US Mailed: August 27, 2003 Sheet 2 of 17
Our Ref. No. 3382-64896

FIG. 3

300

```
abstract structure Exp ←—— 314
  case Const ←—— 306
    val as Integer
  case Bin ←—— 308
    op as Op
    left as Exp ←—— 302
    right as Exp ←—— 304
  case Let ←—— 310
    name as Name
    def as Exp
    body as Exp
  case Var
    name as Name

enum Op
  Add
  Sub

type Name = String
```

FIG. 4

400

```
structure Exp
  public override ToString() as String?
    match me
      x as Const:
        return ToString(x.val)
      x as Bin:
        return "(" + x.left + ToString(x.op) + x.right + ")"
      x as Let:
        return "(let " + x.name + "=" + x.def + " in " + x.body +
")"
      x as Var:
        return x.name

ToString(o as Op) as String?
  match o
    Add: return "+"
    Sub: return "-"
```

Stephen A. Wight
Klarquist Sparkman, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon 97204
Telephone: 503/226-7391

Our Ref. No. 3382-64896
For: ACCESS DRIVEN FILTERING
Express Mail Label EV339201001US Mailed: August 27, 2003 Sheet 3 of 17

FIG. 5

500

```
Closed(e as Exp) as Boolean
  return Closed(e, {}) ← 510

type BoundedNames = Set of Name ← 512
Closed(e as Exp, ns as BoundedNames) as Boolean
  match e ← 502
    Const(_):  return true ← 504
    Bin(_,l,r): return Closed(l, ns) and Closed(r, ns) ← 518
    Let(n,d,b): return Closed(d, ns) and Closed(b, ns + {n})
    Var(n):    return n in ns ← 506
```

508 514 516

FIG. 6

600

```
Eval(e as Exp) as Integer
  require Closed(e, {})
  return Eval(e, {->})

type Environment = Map of Name to Integer
Eval(e as Exp, env as Environment) as Integer
  require Closed(e, Indices(env))
  match e
    Const(v):   return v
    Bin(o,l,r): return Eval(o, Eval(l, env), Eval(r, env))
    Let(n,d,b): return Eval(b, env + {n -> Eval(d, env)})
    Var(n):     return env(n)

Eval(o as Op, l as Integer, r as Integer) as Integer
  match o
    Add:  return l + r
    Sub:  return l - r
```

FIG. 7

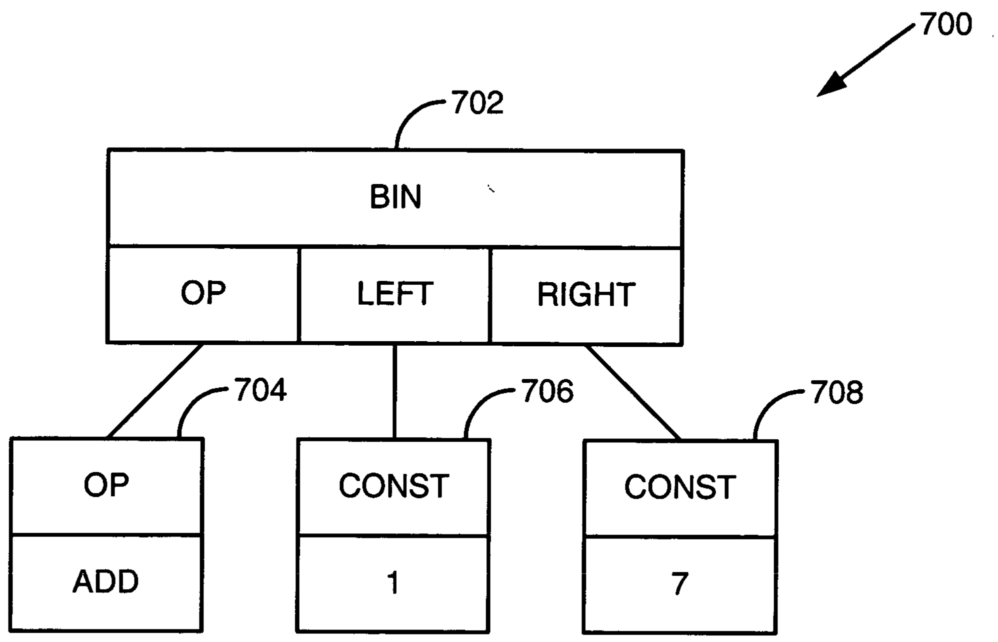

700
702
BIN
OP
LEFT
RIGHT
704
706
708
OP
ADD
CONST
1
CONST
7

FIG. 8

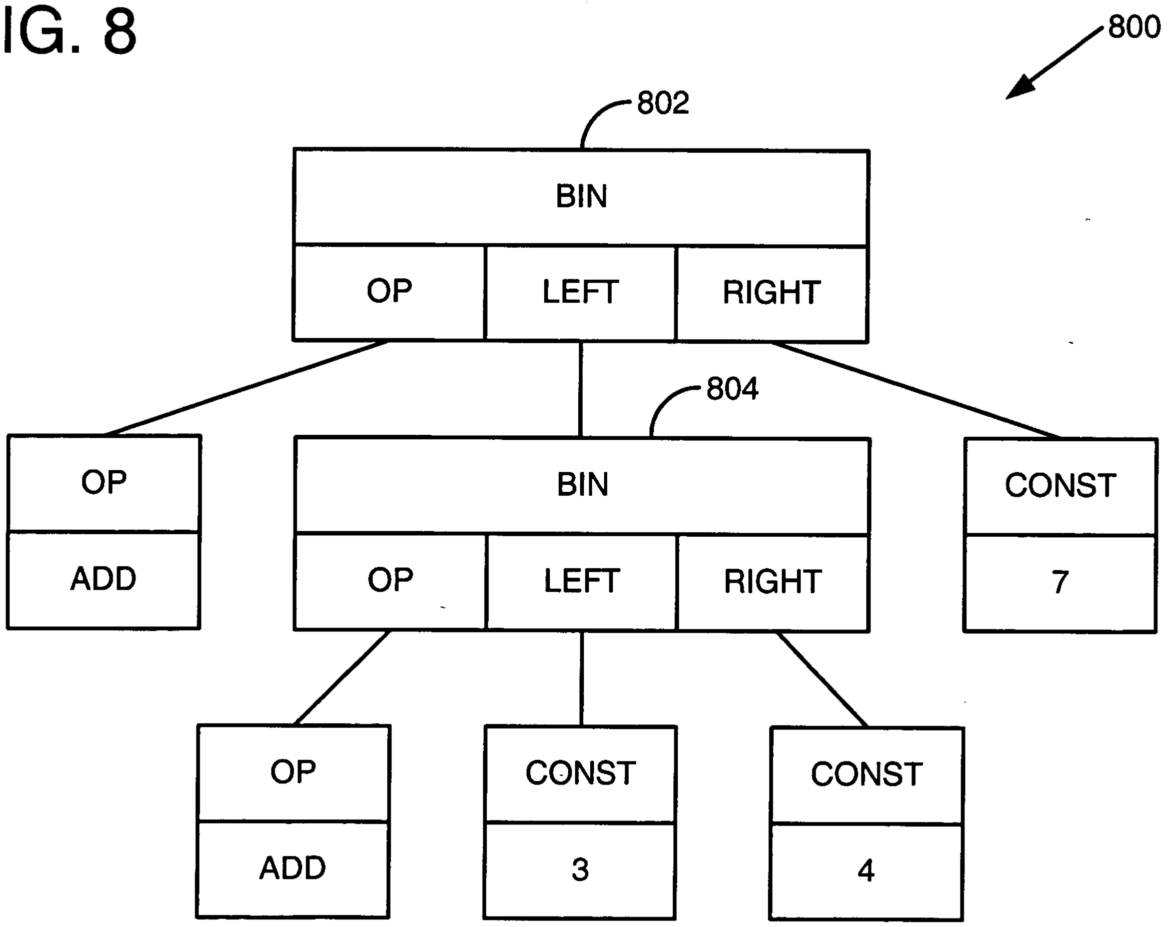

800
802
BIN
OP
LEFT
RIGHT
804
OP
ADD
BIN
OP
LEFT
RIGHT
CONST
7
OP
ADD
CONST
3
CONST
4

FIG. 9

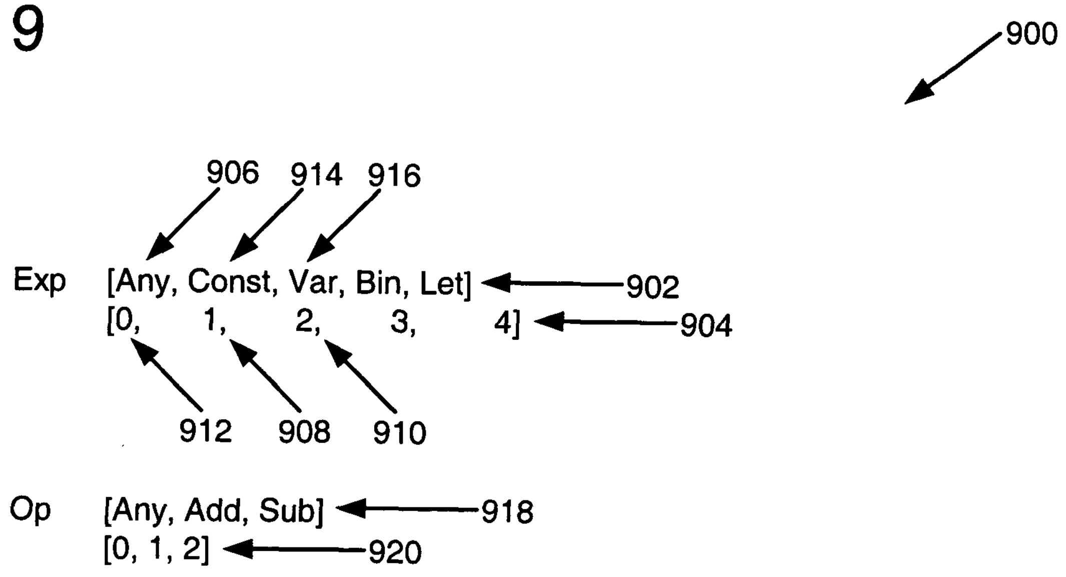
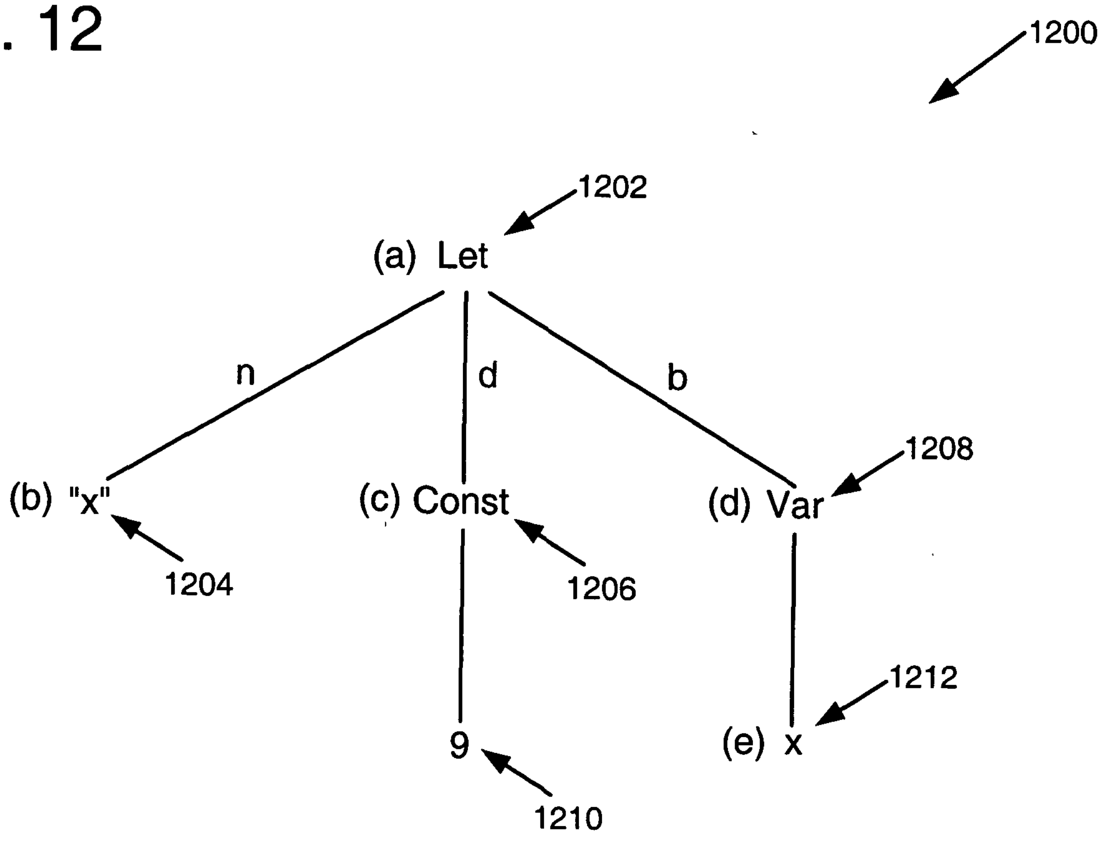
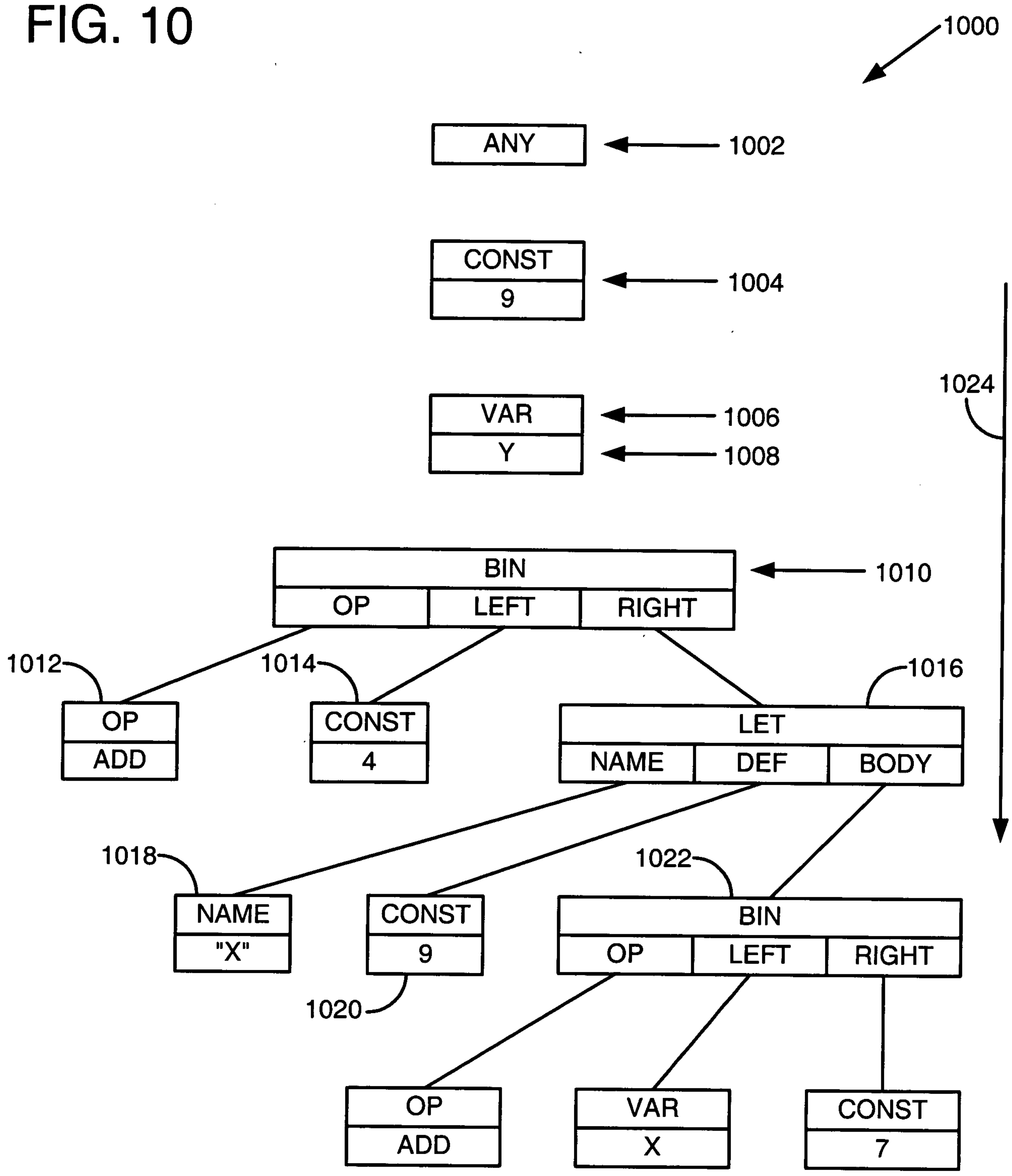

900
906
914
916
Exp
[Any, Const, Var, Bin, Let]
902
[0, 1, 2, 3, 4]
904
912
908
910
Op
[Any, Add, Sub]
918
[0, 1, 2]
920

FIG. 12


1200
1202
(a) Let
n
d
b
(b) "x"
1204
(c) Const
1206
(d) Var
1208
9
1210
(e) x
1212

Stephen A. Wight
Klarquist Sparkman, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon 97204
Telephone: 503/226-7391

FIG. 10



Stephen A. Wight Our Ref. No. 3382-64896
Klarquist Sparkman, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon 97204
Telephone: 503/226-7391
For: ACCESS DRIVEN FILTERING
Express Mail Label EV339201001US Mailed: August 27, 2003 Sheet 7 of 17

FIG. 11

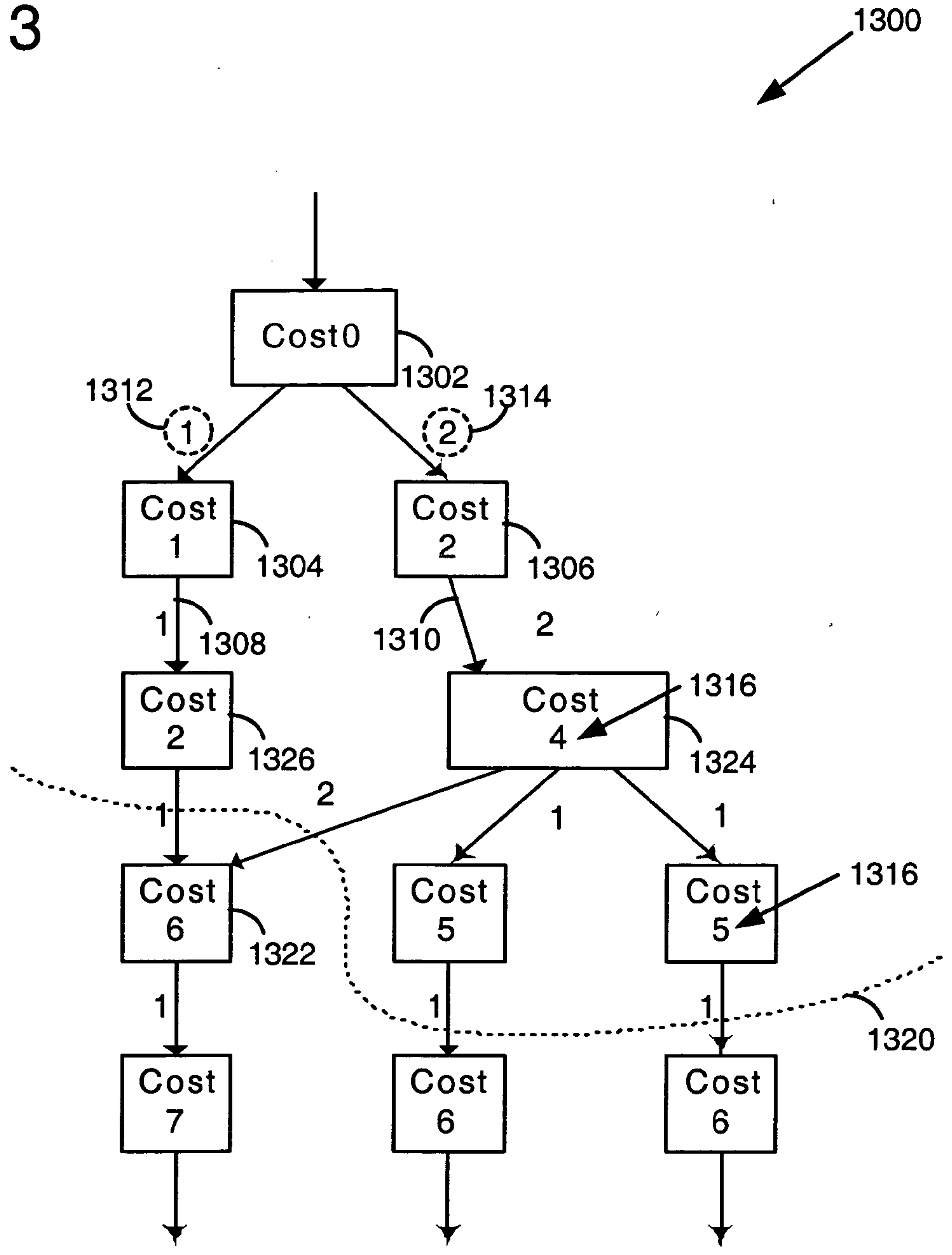
1100

1102

BIN

OP	VALUE	LEFT	VALUE	RIGHT	VALUE
0	ANY	0	ANY	0	ANY
0	ANY	0	ANY	1	CONST
0	ANY	0	ANY	2	VAR
0	ANY	0	ANY	3	BIN
0	ANY	0	ANY	4	LET
0	ANY	1	CONST	0	ANY
0	ANY	1	CONST	1	CONST
0	ANY	1	CONST	2	VAR
0	ANY	1	CONST	3	BIN
0	ANY	1	CONST	4	LET
⋮	⋮	⋮	⋮	⋮	⋮
1	ADD	1	CONST	1	CONST
1	ADD	1	CONST	2	VAR
⋮	⋮	⋮	⋮	⋮	⋮
2	SUB	4	LET	3	BIN
2	SUB	4	LET	4	LET

FIG. 13


1300
Cost0
1302
1312
1
2
1314
Cost 1
1304
Cost 2
1306
1
1308
1310
2
Cost 2
1326
Cost 4
1316
1324
2
1
1
1
Cost 6
1322
Cost 5
Cost 5
1316
1
1
1
1320
Cost 7
Cost 6
Cost 6

Stephen A. Wight
Klarquist Sparkman, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon 97204
Telephone: 503/226-7391

Our Ref. No. 3382-64896

For: ACCESS DRIVEN FILTERING
Express Mail Label EV339201001US Mailed: August 27, 2003 Sheet 9 of 17

FIG. 14

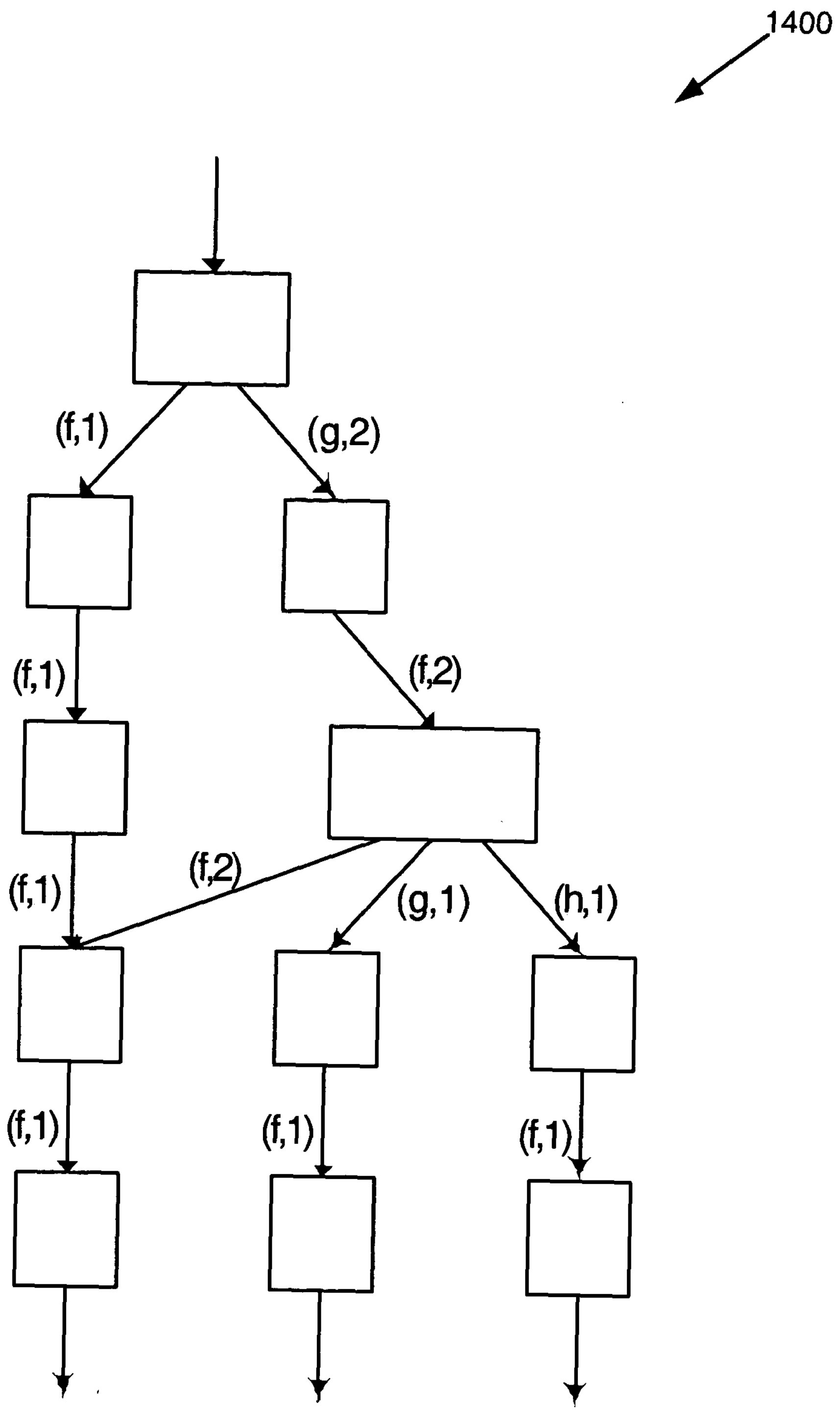
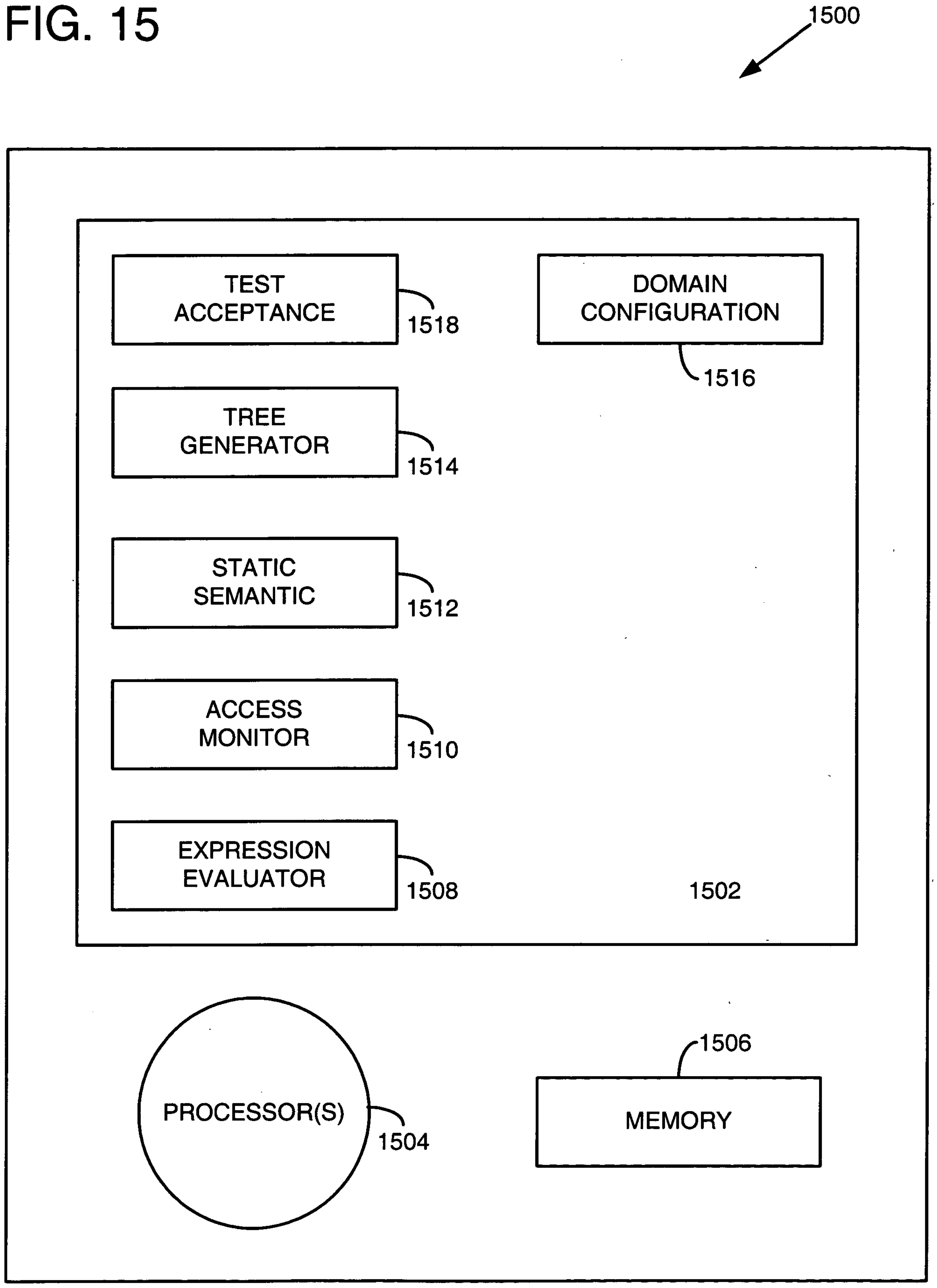


FIG. 15


1500
TEST ACCEPTANCE
1518
DOMAIN CONFIGURATION
1516
TREE GENERATOR
1514
STATIC SEMANTIC
1512
ACCESS MONITOR
1510
EXPRESSION EVALUATOR
1508
1502
1506
PROCESSOR(S)
1504
MEMORY

Stephen A. Wight
Klarquist Sparkman, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon 97204
Telephone: 503/226-7391
Our Ref. No. 3382-64896
For: ACCESS DRIVEN FILTERING
Express Mail Label EV339201001US Mailed: August 27, 2003

FIG. 16

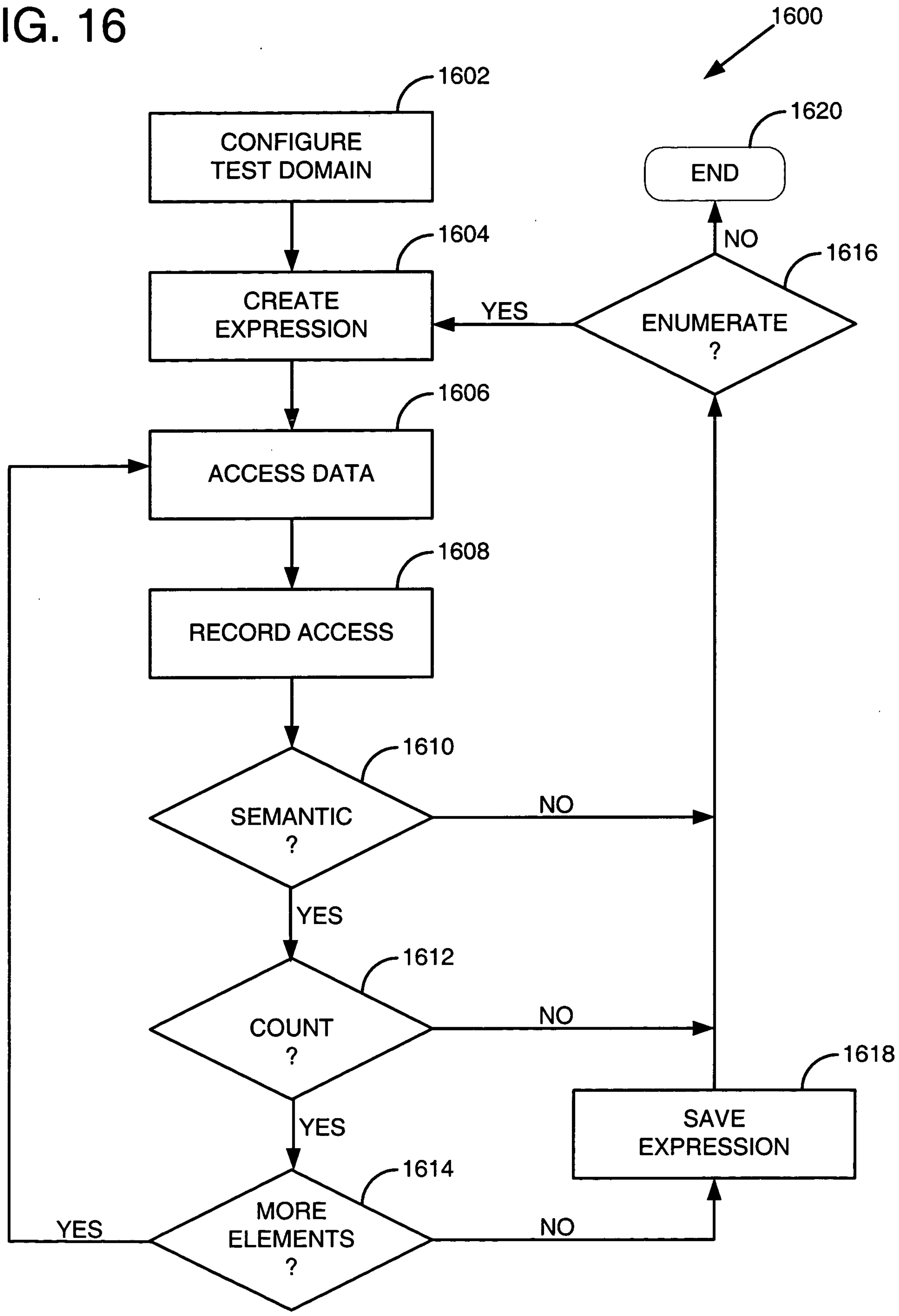
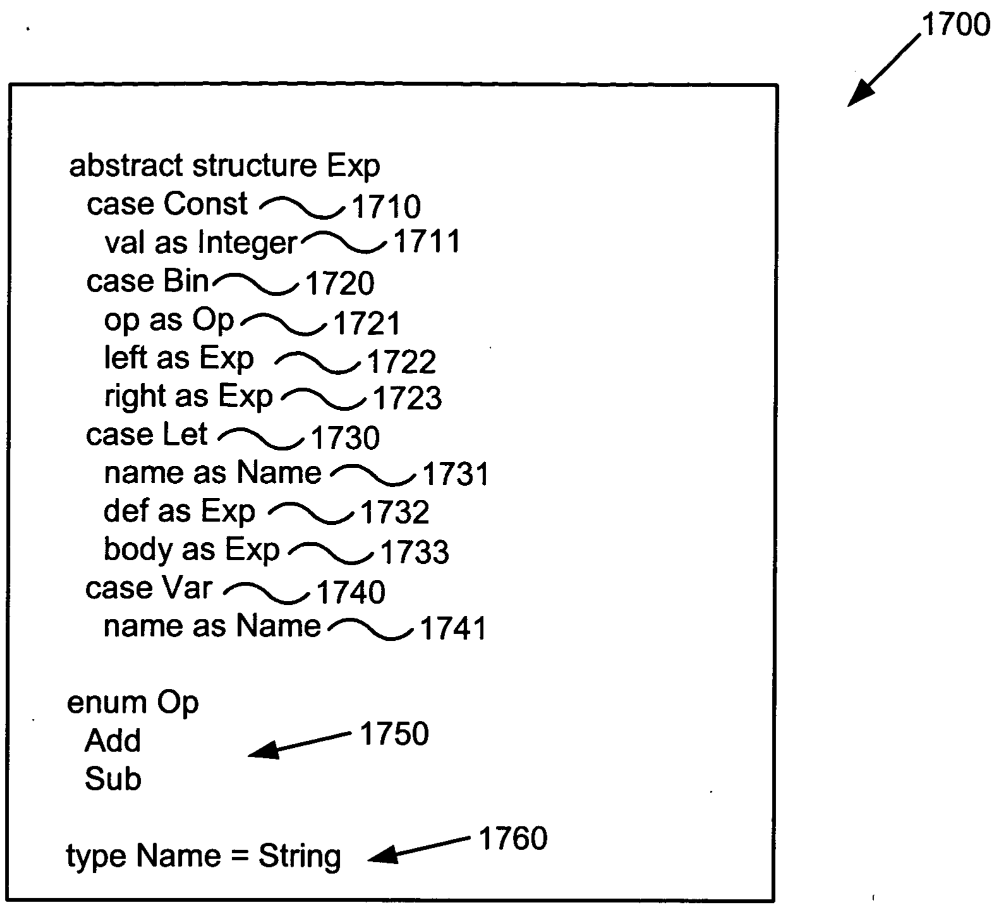


FIG. 17

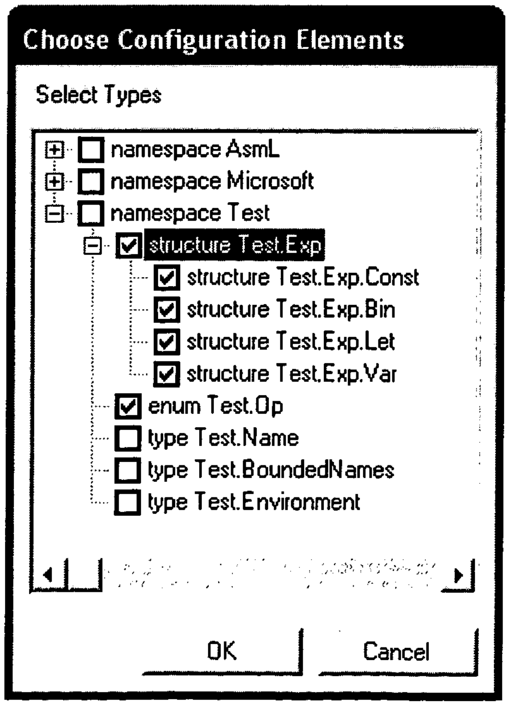


```
abstract structure Exp
  case Const
    val as Integer
  case Bin
    op as Op
    left as Exp
    right as Exp
  case Let
    name as Name
    def as Exp
    body as Exp
  case Var
    name as Name

enum Op
  Add
  Sub

type Name = String
```

FIG. 18

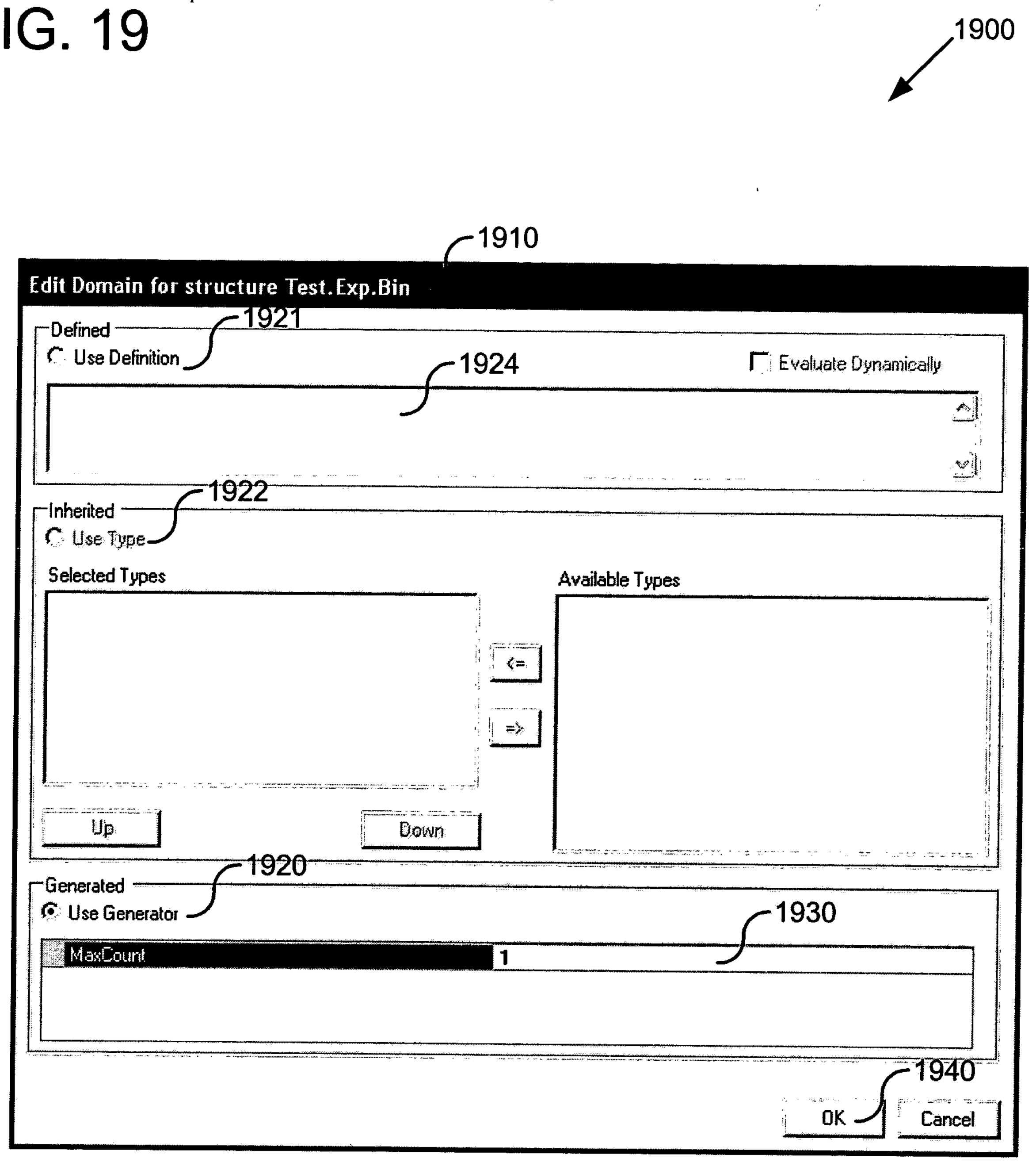


Choose Configuration Elements

Select Types

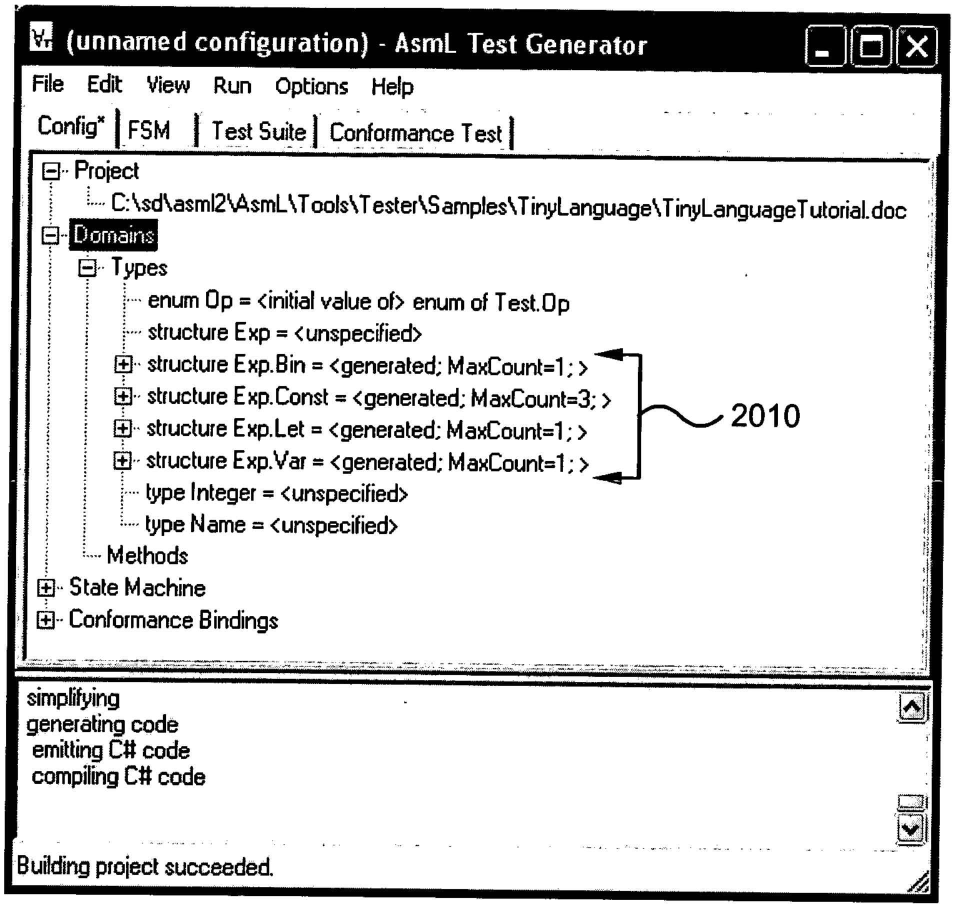
- ☐ namespace AsmL
- ☐ namespace Microsoft
- ☐ namespace Test
  - ☑ structure Test.Exp
    - ☑ structure Test.Exp.Const
    - ☑ structure Test.Exp.Bin
    - ☑ structure Test.Exp.Let
    - ☑ structure Test.Exp.Var
  - ☑ enum Test.Op
  - ☐ type Test.Name
  - ☐ type Test.BoundedNames
  - ☐ type Test.Environment

FIG. 19



MaxCount	1

FIG. 20

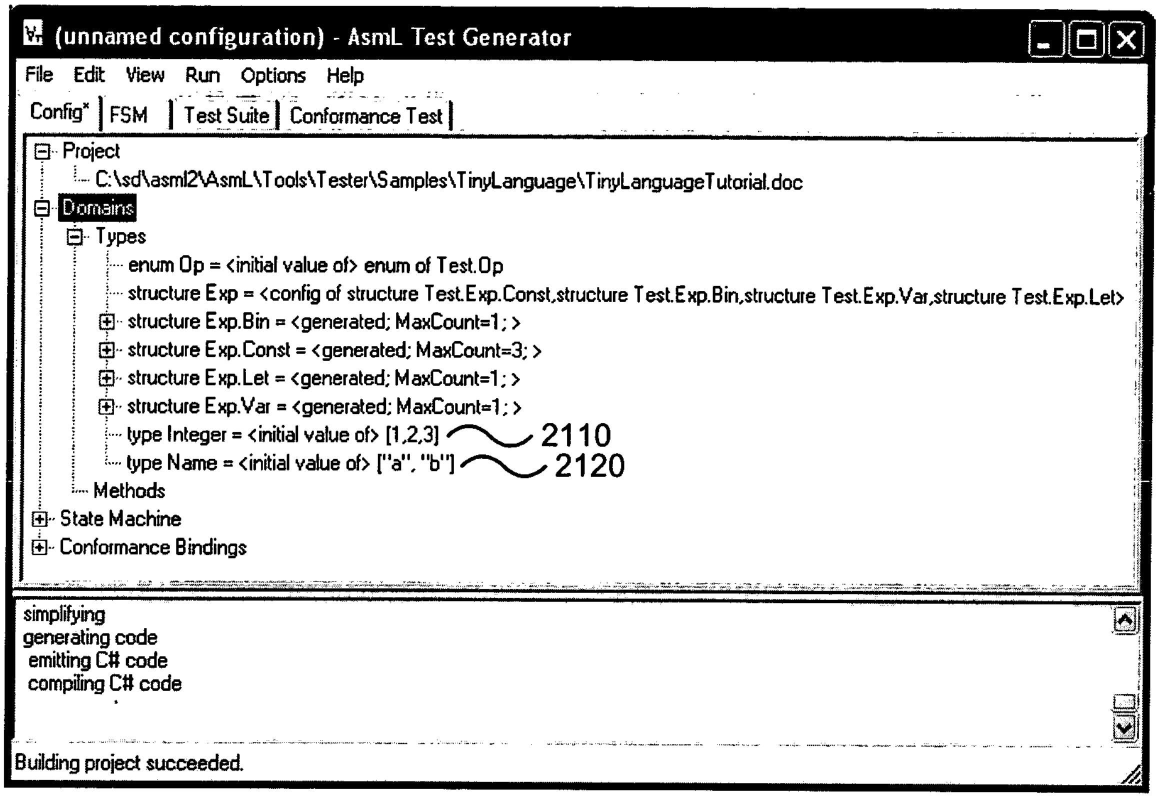

(unnamed configuration) - AsmL Test Generator
File Edit View Run Options Help
Config* FSM Test Suite Conformance Test
2010

```
Project
  C:\sd\asml2\AsmL\Tools\Tester\Samples\TinyLanguage\TinyLanguageTutorial.doc
Domains
  Types
    enum Op = <initial value of> enum of Test.Op
    structure Exp = <unspecified>
    structure Exp.Bin = <generated; MaxCount=1; >
    structure Exp.Const = <generated; MaxCount=3; >
    structure Exp.Let = <generated; MaxCount=1; >
    structure Exp.Var = <generated; MaxCount=1; >
    type Integer = <unspecified>
    type Name = <unspecified>
  Methods
State Machine
Conformance Bindings
```

```
simplifying
generating code
 emitting C# code
 compiling C# code
```

Building project succeeded.

FIG. 21


(unnamed configuration) - AsmL Test Generator
File Edit View Run Options Help
Config* FSM Test Suite Conformance Test

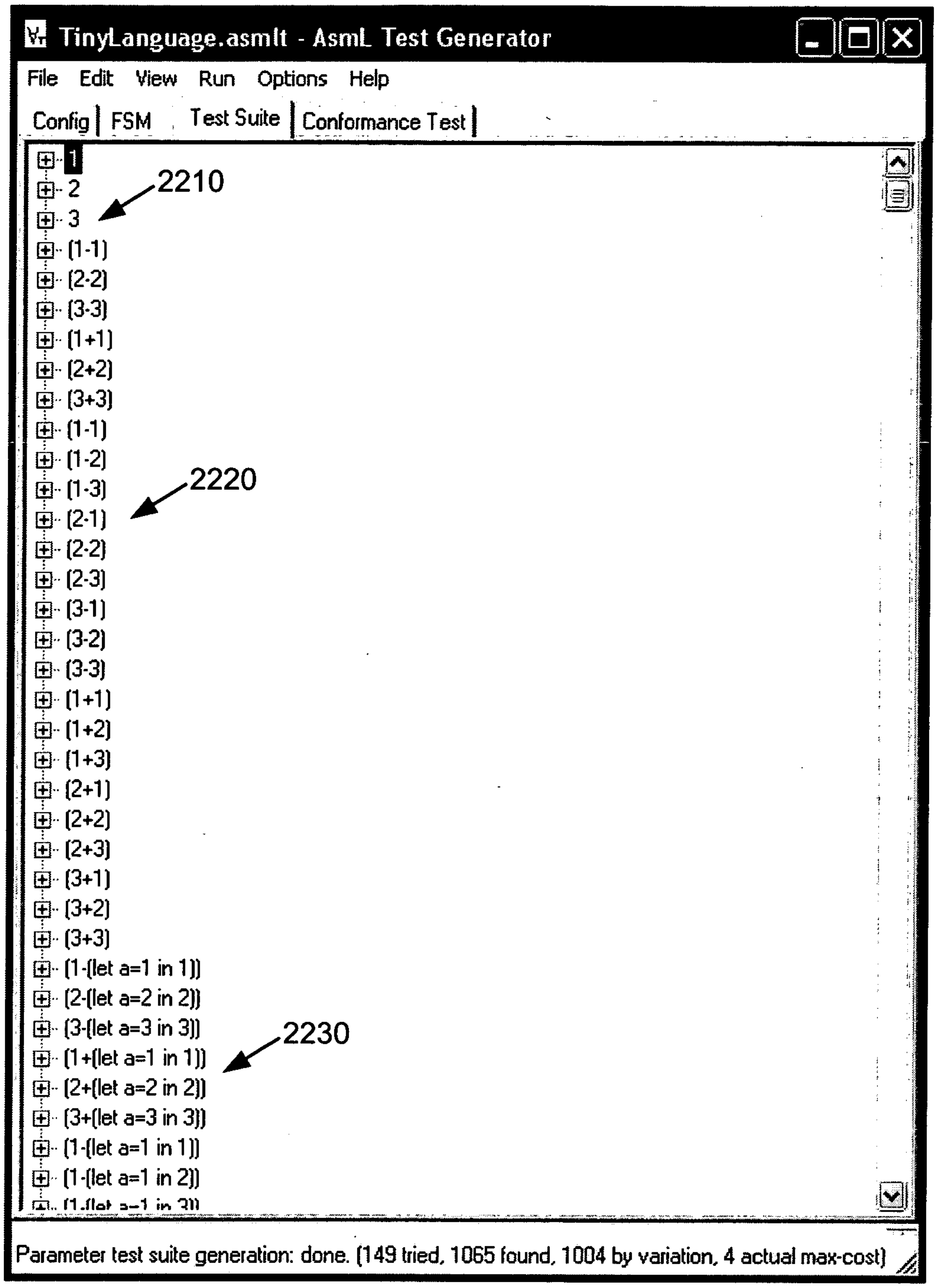
```
Project
  C:\sd\asml2\AsmL\Tools\Tester\Samples\TinyLanguage\TinyLanguageTutorial.doc
Domains
  Types
    enum Op = <initial value of> enum of Test.Op
    structure Exp = <config of structure Test.Exp.Const,structure Test.Exp.Bin,structure Test.Exp.Var,structure Test.Exp.Let>
    structure Exp.Bin = <generated; MaxCount=1; >
    structure Exp.Const = <generated; MaxCount=3; >
    structure Exp.Let = <generated; MaxCount=1; >
    structure Exp.Var = <generated; MaxCount=1; >
    type Integer = <initial value of> [1,2,3]
    type Name = <initial value of> ["a", "b"]
  Methods
State Machine
Conformance Bindings
```

```
simplifying
generating code
 emitting C# code
 compiling C# code
```

Building project succeeded.

Stephen A. Wight
Klarquist Sparkman, LLP
121 SW Salmon Street, Suite 1600
Portland, Oregon 97204
Telephone: 503/226-7391
Our Ref. No. 3382-64896
For: ACCESS DRIVEN FILTERING
Express Mail Label EV339201001US Mailed: August 27, 2003 Sheet 16 of 17

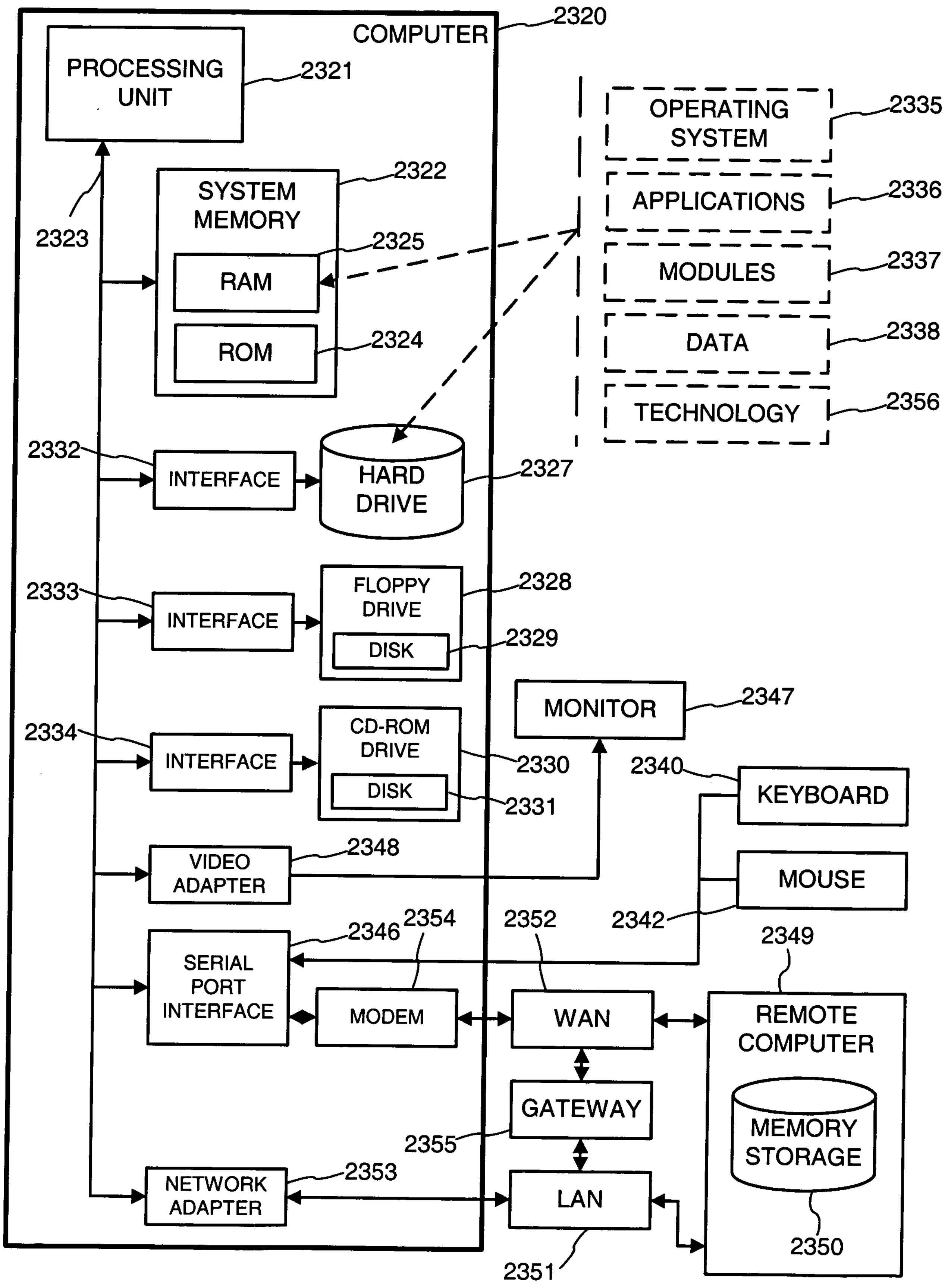
FIG. 22



```
1
2
3
(1-1)
(2-2)
(3-3)
(1+1)
(2+2)
(3+3)
(1-1)
(1-2)
(1-3)
(2-1)
(2-2)
(2-3)
(3-1)
(3-2)
(3-3)
(1+1)
(1+2)
(1+3)
(2+1)
(2+2)
(2+3)
(3+1)
(3+2)
(3+3)
(1-(let a=1 in 1))
(2-(let a=2 in 2))
(3-(let a=3 in 3))
(1+(let a=1 in 1))
(2+(let a=2 in 2))
(3+(let a=3 in 3))
(1-(let a=1 in 1))
(1-(let a=1 in 2))
(1-(let a=1 in 3))
```

Parameter test suite generation: done. (149 tried, 1065 found, 1004 by variation, 4 actual max-cost)

FIG. 23


COMPUTER
2320
PROCESSING UNIT
2321
SYSTEM MEMORY
2322
2323
RAM
2325
ROM
2324
OPERATING SYSTEM
2335
APPLICATIONS
2336
MODULES
2337
DATA
2338
TECHNOLOGY
2356
2332
INTERFACE
HARD DRIVE
2327
2333
INTERFACE
FLOPPY DRIVE
2328
DISK
2329
2334
INTERFACE
CD-ROM DRIVE
2330
DISK
2331
MONITOR
2347
2340
KEYBOARD
VIDEO ADAPTER
2348
MOUSE
2342
SERIAL PORT INTERFACE
2346
2354
MODEM
2352
WAN
2349
REMOTE COMPUTER
MEMORY STORAGE
2350
GATEWAY
2355
NETWORK ADAPTER
2353
LAN
2351